

Formulas and Functions

CHAPTER 4

Formulas and Functions

Chapter 4 describes the types of formulas and functions available for computations with SuperCalc3.

SuperCalc3 Formulas

Formulas specify mathematical calculations and relationships. They consist of operands and operators combined in such a way as to produce a value. When entered into a cell, a formula becomes the cell *content*.

A maximum of 116 characters can be entered into any one cell. You can construct a longer formula by entering parts of it into separate cells, then referencing those cells.

Formula Values

The value of a cell containing a formula is the result obtained by evaluating the content in the cell. A formula may take on five types of values.

- Numeric
- Date
- Textual
- Not Available
- Error

The type of value that a formula may compute is not fixed when the formula is entered. This is in contrast to the cell type that is fixed to the formula when the cell content is non-empty and does not begin with a (") or (') character.

To illuminate this difference, consider the following example. This formula defines a formula cell (Form =), but the dynamic value type is determined by the value of cell A1 and may be any of the five possible types of values.

IF(A1 = 1,NA,IF(A1 = 2,ERROR,IF(A1 = 3,PI,IF(A1 = 4,("Textual"),TODAY))))

This expression evaluates:

If A1 = 1, the value is the Not Available value.

If A1 = 2, the value is the Error value.

If A1 = 3, the value is the numeric value 3.14159265358979

If A1 = 4, the value is Textual.

If A1 = 5, the value is the Date value representing today.

Operators

SuperCalc3 uses three types of mathematical operators.

Arithmetic Operator

An arithmetic operator defines the arithmetic operation performed between two numeric operands. The SuperCalc3 operators are described below.

+	Addition
–	Subtraction
*	Multiplication
/	Division
%	PerCent
^ or * *	Exponent

The arithmetic operators are evaluated according to algebraic precedence. The exponent operator is evaluated first. The multiplication, division and percent operators are evaluated next. The addition and subtraction operators are evaluated last.

Examples:

1) 4 + 5 * 2 ^ 2 is the same as
 4 + (5*(2 ^ 2)) or
 24

2) -2 ^ 2 is the same as
 -(2 ^ 2) or
 -4

Relational Operators

A relational operator compares two operands and returns a value of true or false. A true comparison has a numerical value of 1, false a numerical value of 0.

The following sample compares terms *a* and *b* using the relational operators:

a = b Equal: The relation is true (1) if, and only if, *a* is equal to *b*. All other cases are false (0).

a < >b Not Equal: The relation is true (1) if *a* does not equal *b*. All other cases are false (0).

a < b Less Than: The relation is true (1) if the value of *a* is less than the value of *b*. It is false (0) if the value of *a* is greater than or equal to *b*.

a > b Greater Than: The relation is true (1) if the value of *a* is greater than the value of *b*. It is false (0) if the value of *a* is less than or equal to *b*.

a < = b Less Than or Equal To: The relation is true (1) if the value of *a* is less than or equal to the value of *b*, It is false (0) if the value of *a* is greater than the value of *b*.

a > = b Greater Than or Equal To: The relation is true (1) if the value of *a* is greater than or equal to the value of *b*. It is false (0) if the value of *a* is less than the value of *b*.

The relational operators = and < > can be used to compare any of the five types of values. The other relational operators <, >, <=, >= can be used to compare numeric and date values only.

Parentheses Operators

Parentheses operators define the precedence order of calculation within a mathematical formula. Operations enclosed within parentheses are calculated first. The use of parentheses overrides the algebraic precedence order of arithmetic operators. Parentheses can be nested.

Operands

An operand is a numerical value. It may be obtained as the result of a constant, a cell reference, the evaluation of a formula, or function.

Constants

There are two types of constants: numeric and textual.

Numeric Constant (Value)

A numeric constant is any number such as an integer or decimal number or an exponential number. SuperCalc3 accepts a maximum 16 decimal places for a numeric constant.

Textual Constant (Value)

SuperCalc3 allows you to enter text (non-numeric characters) into a cell and subsequently reference the cell content in a formula expression. Enter the text as a textual value by enclosing it in double quotes and parentheses. For example, to enter the word Debit as a textual value, enter **("Debit")**.

This is quite different from text entered as a Text Cell. Such text has a value of zero when referenced in a SuperCalc3 formula.

Text entered as a textual value may be referenced by other cells either singly or used to construct certain expressions. Such references may be used in the construction of lookup tables and conditional expressions.

A textual value has the following characteristics:

- A maximum of nine characters are accepted. If you exceed this limit, the entry is accepted as text.
- A textual value may contain any character including punctuation characters and numbers. Numbers in a textual value do not have any mathematical significance.
- Use three pairs of double quote characters if you want to enter a quote mark as part of the textual value. For example:

 (""""SELL"""") produces "SELL"

- A textual value is similar to a standard text entry except that a textual value can be propagated. That is, the value may be referenced by other cells. Because of this, SuperCalc3 considers them to be formulas.

Cell References

The value of a cell may be used as an operand by naming the coordinates of that cell in a formula.

Functions

A SuperCalc3 function returns the value of a calculation. There are five types of SuperCalc3 functions:

- Arithmetic
- Logical
- Calendar
- Financial
- Special

To use one of these functions, you enter its name, possibly followed by arguments. The arguments specify the values that you want to apply to the function.

Arithmetic Functions

The following are the arithmetic functions. An argument may consist of a value, a range, or a list.

- Value — An expression evaluating to a numeric value.
- Range — A group of cells specified by naming the top left-most cell and the bottom right-most cell, separated by a colon.
- List — ranges and values separated by commas.

ABS(Value)

Returns the absolute value of the Value given.

- Equivalent to the value itself if positive.
- Equivalent to the value without its negative sign if negative. This is the additive inverse.
- Equivalent to Zero if the expression is zero.

Example: **ABS(-237)** = 237

ACOS(Value)

Returns the radian angle of the cosine value given.

Example: **ACOS(1)** = 0

ASIN(Value)

Returns the radian angle of the sine value given.

Example: **ASIN(.2)** = .2013579207903336

ATAN(Value)

Returns the radian angle of the tangent value given.

Example: **ATAN(2)** = 1.107148717794091

AVERAGE or **AV (List)**

Returns the average (mean) of the range given. This function is equivalent to the SUM of the list divided by the COUNT of the list.

Example: **AVERAGE(H2:H20)**

COS(Value)

Returns the cosine of the radian angle value given.

Example: **COS(PI)** = -1

COUNT(List)

Returns the number of non-blank non-text cells described by the range.

Example: **COUNT(H2:H20)** = 18 if the list is full.

EXP(Value)

This function raises the number e exponentially to the *value*. The value of e is 2.71828182845904.

Example: **EXP(2)** e ^ 2 or 7.3890560989306

INT(Value)

Returns the integer of the value given, the value is not rounded.

Example: **INT(2.5832)** = 2

LN(Value)

Returns the natural log, log base e, of the value given.

Example: **LN(5)** = 1.609437912434

LOG10 or **LOG(Value)**

Returns the common log, log base 10 of the value given.

Example: **LOG10(12)** = 1.07918124604759

MAX(List)

Returns the maximum value of the range. Non-numeric cells are ignored.

Example: **MAX(A1:A20)**

MIN(List)

Returns the minimum value of the range. Non-numeric cells are ignored.

MOD(value1,value2)

The MOD function produces the remainder that results from the division of 'value1' by 'value2'.

MOD(10,7) produces 3

The remainder when dividing 10 by 7 is 3.

PI

Returns the value of Pi to 15 significant digits.

Example: **PI** = 3.14159265358979

RANDOM or RAN

Returns a random number between 0-1.

Example: **RAN** = .6913285697213548

To get a whole number use the INT function

Example: **INT (RAN*10e3)** = 691

Note that the power of 10 by which you multiply RAN gives you the number of digits in your whole number.

ROUND(Value,Places-Value)

Use ROUND to round a value to a specified number of places. First specify the value to be rounded, then the place holder where the rounding is to occur. Use – to designate positions to the left of the decimal and + to designate positions to the right of the decimal. The + sign is optional; if it is omitted, a positive number is assumed. For example:

ROUND(1234.5678,2) = 1234.57

ROUND(1234.5678,-2) = 1200

SIN(Value)

Returns the sine of the radian angle value given.

Example: **SIN(PI/2)** = 1

SQRT(Value)

Returns the square root of the value.

Example: **SQRT(4)** = 2

SUM(List)

Returns the sum of the values in the range. Non-numeric cells are ignored.

Example: **SUM(A4,B15,C15:C20)**

TAN(Value)

Returns the tangent of the radian angle value given.

Example: **TAN(.75*PI)** = -1

Special Considerations:

- A formula may be used to produce a value. SuperCalc3 evaluates the formula and uses the value for the argument to the function.
- Specify a range of cells for Range.

Logical Functions

A logical function consists of a relational comparison connected by a logical operator. Complex logical expressions may be formed by using parentheses.

IF(Expression1,Value2,Value3)

If expression 1 is true, enter value 2 into the cell. If expression 1 is false, enter value 3.

If an expression is entered into an IF function, the expression must evaluate properly to a value in order for the IF function to be valid. That is, the expression must not result in a Formula ERROR.

Value 3 may be omitted. In that case, the value of the expression is zero if value 1 is false.

AND(Value1,Value2)

A logical *AND* function has a value of true (numerical value of 1) if both value 1 and value 2 are true. If either value is false, the AND function is false (numerical value of 0).

AND(True,True)	True
AND(True,False)	False
AND(False,True)	False
AND(False,False)	False

Example: **AND(H6 = 5,B3 <> 8)**

Returns the value of 1 if both conditions are true. Returns the value of 0 if either condition is false.

OR(Value 1,Value 2)

A logical *OR* function has a value of true (numerical value of 1) if either value 1 or value 2 is true. If both values are false, the OR function is false (numerical value of 0).

OR(True,True)	True
OR(True,False)	True
OR(False,True)	True
OR(False,False)	False

Example: **OR(B1 >= 74.2,C3 = 3)**

Returns the value of 1 if one or both values are true. Returns the value of 0 if both values are false.

NOT(Value)

The NOT function returns the opposite truth value as the stated value.

NOT (True)	False
NOT (False)	True

Example: **NOT(B5 >= 5.9)**

Returns the value of 1 if the value is false. Returns the value of 0 if the value is true.

Additional Examples:

IF functions are easy to work with when you remember these few simple pointers.

1. IF Functions look like this:

 IF(Expression A,Expression B,Expression C)

2. They read as follows:

 If Expression A is true, then use Expression B.

 If Expression A is false, then use Expression C.

3. In other words:

 If Expression A, then Expression B, otherwise, Expression C.

Consider the IF function:

IF(A1 >=5000,10,5)

If the cell A1 contained the value 455, the cell this formula resided in would show the value 5.

Now, suppose that you need to evaluate two IF functions at the same time. Consider this example:

IF(A1 =5000,5,IF(A1 =4000,25,0))

Notice that the IF function still reads *If Expression A, then Expression B, otherwise Expression C*. It just happens that Expression C is another IF function.

Expression B or Expression C can be a formula or another IF function. You can continue to build your formula up to 116 characters.

Note: There must always be as many closed parentheses as there are open. This is important.

Let's look at two more analogies that may also be useful.

IF-AND Combinations

1. IF-AND combinations look like this:

 IF(AND(Exp Aa,Exp Ab),Exp B,Exp C)

2. They read as follows:

 If Aa and Ab are both true, then use Expression B. If either Aa or Ab is false, then use Expression C.

3. In other words:

 If Expression Aa and Ab are both true, then Expression B, otherwise Expression C.

 Example:

 IF(AND(A1 > 500,A1 < 1000),5,0)

 Both functions in Expression Aa and Ab must be true in order to evaluate Expression B.

IF-OR Combinations

1. IF-OR combinations look like this:

 IF(OR(Exp Aa,Exp Ab),Exp B,Exp C)

2. They read as follows:

 If either Aa or Ab are true, then use Expression B. If Aa and Ab are both false, then use Expression C.

 Example:

 IF(OR(A1 > 5000,B1 < 100),5,0)

Only one of the functions, Expression Aa or Ab has to be true in order to use Expression B.

The SuperCalc3 Calendar Functions:

SuperCalc3 features a calendar that allows you to enter a date into your spreadsheet, then reference that date in calculations for other cells.

SuperCalc3 uses a Modified Julian Calendar that ranges from March 1, 1900 to February 28, 2100. Days in this 200 year range are numbered sequentially from 1 through 73049.

The SuperCalc3 calendar functions fall into two categories: (1) Date Entry functions and (2) Date Reference functions.

SuperCalc3 displays a date according to the conventional format MM/DD/YY. Although expressed using numbers, it does not constitute a numeric entry. A date value is a special value and can only be referenced by the Date Reference functions. The other functions of SuperCalc3 treat the Data as a text entry; i.e., it has a numerical value of 0.

Date values can be used with some arithmetic operations.

1. You can add a number to a date with the result being a date value.

 Example: If Cell A1 has the date value 3/13/83. The formula **A1 + 45** produces the date value 4/27/83.

2. You can subtract a number from a date with the result being a date value.

Date Entry Functions:

You enter a Date into your spreadsheet using the Date Entry Functions: Date (or DAT), TODAY, or DVAL.

DATE or **DAT(MM,DD,YY)**
DATE or **DAT(MM,DD,YYYY)**

Enter the month, day and year in that order separated by commas. The year may be entered either as a two digit or four digit number. SuperCalc3 assumes two digit years are 20th Century and adds 1900 to the entry. You must enter a 21st Century date using 4 digits.

You may enter single digit values without a leading 0. For example, the date for February 8, 1905 could be entered as:

DATE(2,8,5)

For the default column width (9), SuperCalc3 will display the date showing only the last 2 digits of the year, even if you enter a 4 digit value. There is no distinction on screen between centuries, even though SuperCalc3 keeps track of them internally. The column width must be at least 11 to see all four digits. Then, you will see a 4 digit year, even if you enter it as a 2 digit number.

Only those dates within the 200 year range of the SuperCalc3 calendar are accepted. If you attempt to enter a date that is not valid, a Formula ERROR will result.

TODAY

The TODAY function reads the system date into the current cell. The desired date must be previously entered into your system. If not, the default system date displays in the cell (or N/A with some systems).

A disk file containing a TODAY cell looks for the current system date when loaded into your spreadsheet. Of course, any other cells that reference a TODAY cell will be evaluated based on the current date. If you don't want the date to be dependent on the system date, use the DATE function.

DVAL(Value)

The DVAL function returns the date of the value specified. The value must be an integer between 1 and 73049. DVAL is the inverse of JDATE.

Date Reference Functions:

SuperCalc3 contains five Date Reference functions. You specify the function first, followed by the address of the cell containing the date value, or a Date Entry function such as TODAY. SuperCalc3 places the formula in the current cell and returns the value. The referenced cell must contain a valid DATE or an ERROR will be indicated. This value can be used as any other value.

Note: The Date Reference functions return normal numeric values. See *Examples* section below.

MONTH or MON(Date Value)

The MONTH function returns the number of the month of the date value (1 for January, 12 for December).

DAY(Date Value)

The DAY function returns the number of the day of the month of the date value.

YEAR(Date Value)

The YEAR function returns the number of the year of the date value.

WDAY(Date Value)

The WDAY function returns the Julian number of the day of the week of the date value (1 for Sunday, 7 for Saturday).

JDATE(Date Value)

The JDATE function returns the Modified Julian Date of the date value. This number ranges from 1 (March 1, 1900) through 73049 (February 28, 2100).

Examples:

A (*Date Value*) entry might be: (*TODAY*); or (*B1*) where cell B1 contains a date entered with a Date Entry function; or (DATE(9,23,84)); or (DVAL(2000)).

In the following examples, assume cell A3 contains the system date 8/16/84:

MON(A3) or **MON(TODAY)** returns *8*.
DAY(A3) or **DAY(TODAY)** returns *16*.
YEAR(A3) or **YEAR(TODAY)** returns *1984*.
WDAY(A3) or **WDAY(TODAY)** returns *5* (for Thursday).
JDATE(A3) or **JDATE(TODAY)** returns *30850*.

Special Considerations:

1. Lookup tables may be used to *convert* the numeric value of the Date Reference functions to their corresponding names. Be sure to specify the names using the textual value format. For example, to convert WDAY functions to the day of the week, set up a lookup table as follows:

 1 ("Sunday")
 2 ("Monday")
 ...
 7 ("Saturday")

 Then, using the WDAY function, you can print the week day for the date. For example: Lookup (WDAY(B1),Z1:Z7), where B1 = Today, and Z1- Z7 is the range of day numbers shown above.

2. You can perform two types of calculations using Date values.

 A. You may add a numeric value to (or subtract the value from) a date. The number is assumed to represent days and the result produces a new date. For example:

 12/25/82 + 7 produces 1/ 1/83

 2/25/86 – 365 produces 2/25/85

B. You may subtract one date from another. The result is expressed as a numeral representing the number of days separating the two dates.

10/31/83 – 7/4/83 produces 119

Note: A number minus a date produces an ERROR.

3. SuperCalc3 accepts numbers outside the legitimate range of dates. Such dates are converted to their legal counterparts if possible. If this is not possible, a Formula ERROR results.

Example: **DATE(6,60,83)** is converted to 7/30/83

Example: **DATE(15,01,82)** is converted to 3/1/83

This feature allows you to conveniently create dates that span logical new months or years using the **R**eplicate command.

Financial Functions

The SuperCalc3 program includes five financial functions.

NPV(Discount,Col/Row Range)

Returns the present value of a group of cash returns at the given rate of discount (for example, a discount rate of 10% would be entered as .10). The cash amounts are assumed to be projected for equal time periods, such as yearly, and the discount rate is for that interval. The first cash entry is discounted once, the second twice and so forth and added to the total value. Net Present Value is the present value of future cash flows, discounted at the appropriate cost of capital, minus the cost of the investment.

For example, with an initial investment of $-10,000 (Cell A1) and returns of 2000, 2400, 2800, 3450 and 2800 in cells B1 through F1 and a discount rate of 8%, calculate net present value in cell A2 as
A2 = NPV (.08,B1:F1) + A1, which would yield NPV = 573.

$$NPV = \sum_{j=1}^{k} A_j \, (1 + r)^{-j}$$

j = Period number (from 1 to k)
Aj = Cash flow at period
r = Rate of interest (discount rate)

IRR(Guess,Row/Col Range)

The Internal Rate of Return is the discount rate such that the Net Present Value of the cash flow is zero.

Guess	Is the initial value to use as the discount rate. This is an optional parameter which can be specified either as a cell address or as a value. If you do not specify an initial value for the discount rate, the program assumes a default value of 0.1.
Row/Col Range	A row or column range where the first element of the range is the initial investment and the remaining entries are the cash flows at regular intervals.
Note:	Usually, an initial value between 0.0 and 1.0 will yield a result.

An iterative technique is used to compute the IRR. This function returns ERROR if convergence to within 0.0000001 does not occur within 20 iterations. In this case, repeating this computation with a different value for the guess may produce a valid result.

Example: With an initial investment of $-10,000 and returns of 2000, 2400, 2800, 3450, and 2800 in cells A1:A6, and an initial guess of .01, in cell B1, the IRR is computed as follows:

IRR(B1,A1:A6) = .1000113 (10.00113%)
or
IRR(.01,A1:A6) = .1000113

Notes: In some cases IRR may have a multiple valued solution. In general, only one interest rate exists for any series of cash flows where the zero period cash flow is the ONLY negative payment or where the negative cash flows are in sequence beginning with period zero.

Where cash flows have multiple negative payments not in sequence, multiple interest rates may exist. In such cases, different initial guesses may result in different answers.

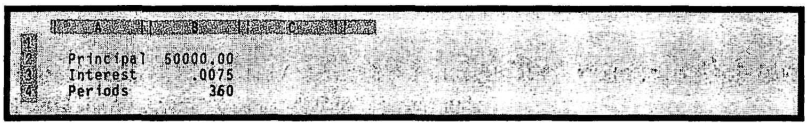
PMT(Principal,Interest rate,Periods)

This function calculates the Payment per period for an ordinary annuity given the Principal, Interest Rate per period and the number of Periods. The formula used for an ordinary annuity is:

$$PMT = PV * \frac{i}{1 - (1 + i)^{-n}}$$

where i = interest rate per compounding period
n = number of periods
PV = principal or present value at the beginning of period 1
and PMT = payment made at the end of each period

Example: The monthly payment on a $50,000 mortgage at 9% annual (.0075 monthly) interest for a 30 year (360 months) term is calculated in the following screen:



PMT(50000,.0075,360) = 402.31
or
PMT(B2,B3,B4) = 402.31

FV(Payment,Interest rate,Periods)

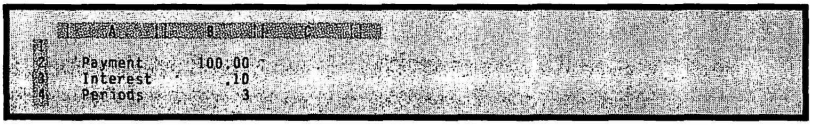
This function calculates the Future Value of equal payments involved in a simple ordinary annuity, given the interest rate per period and the number of periods.

The formula used is:

$$FV = PMT * \frac{(1 + i)^n - 1}{i}$$

where i = interest rate per compounding period
n = number of periods
PMT = constant cash payment each period
and FV = amount at the end of "n" periods

Example: The value at the end of the third year of $100 invested at the end of each year for 3 years at a 10% annual interest rate is computed in the following screen:



	A	B	C
1			
2	Payment	100.00	
3	Interest	.10	
4	Periods	3	

FV(100,.1,3) = 331
or
FV(B2,B3,B4) = 331

PV(Payment,Interest rate,Periods)

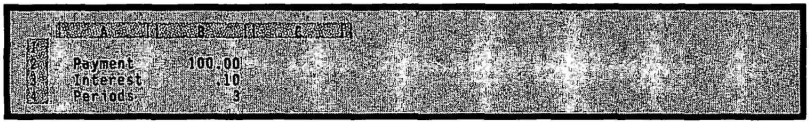
This function calculates the Present Value of equal payments involved in a simple ordinary annuity, given the interest rate per period and the number of periods.

The formula used is:

$$\mathbf{PV} = \mathbf{PMT} * \frac{1 - (1 + i)^{-n}}{i}$$

where i = interest rate per compounding period
n = number of periods
PMT = payment made at the end of each period
and PV = present value at the beginning of period 1

Example: The present value of $100 received at the end of each year for 3 years if the payments are discounted at an annual rate of 10% is computed as follows. (This example assumes a **$** format setting):



	A	B	C
1			
2	Payment	100.00	
3	Interest	.10	
4	Periods	3	

PV(100,.10,3) = 248.69
or
PV(B2,B3,B4) = 248.69

Special Functions

The program includes a number of Special Functions.

ERROR or **ERR**

Displays *ERROR* in a cell that returns a value that cannot be calculated. You can enter the term ERROR into a cell by typing it on the data entry line.

LOOKUP or **LU(Value, Col/Row Range)**

Searches for the last value in the range of numbers that is less than or equal to the search value given and returns the adjacent value to the right of the search column or below the search row. This function assumes that the search range is in ascending order of values.

A lookup table consists of two adjacent rows or columns containing data. A lookup table can be either horizontal or vertical. SuperCalc3 searches the left column of a vertical lookup table and returns the adjacent value in the right column. SuperCalc3 searches the top row of a horizontal lookup table and returns the adjacent value in the bottom row.

Note: Text strings cannot be *looked up* in a lookup table. Only values can be looked up. To look up text, enter it as a textual value, for example, **("Monday")**.

NA

Displays N/A in a cell for which data are not available. You can enter the value N/A into a cell using **NA**.

Note: You enter **NA** (without a slash) and SuperCalc3 displays N/A (with a slash).

The following five functions test for cell type. They return true (1) or false (0).

ISNUM(Cell)

Tests the cell for a numeric value type. Note that text and repeating text have a numeric value of zero, thus evaluate as true.

ISDATE(Cell)

Tests the cell for a date value type.

ISTEXT(Cell)

Tests the cell for a textual value type. Note that this is not a test for a text or repeating text cell.

ISNA(Cell)

Tests the cell for a Not Available value type.

ISERROR(Cell)

Tests the cell for an Error value type.

Examples:

IF(ISERROR(A14),Expression2,Expression3)

If the contents of A14 are ERROR, then use expression 2, otherwise use expression 3.

IF(ISDATE(C33),Expression2,Expression3)

If the contents of C33 are a Date Value, then use expression 2, otherwise use expression 3.

TRUE

True evaluates to the numeric value 1.

FALSE

False evaluates to the numeric value 0.

Special Considerations:

Textual Values in Formulas and Functions

A textual value is used like other operands in the construction of expressions for lookup tables (the LOOKUP function) and logical functions. Due to the nature of a textual value however, it may not be used in some situations where a numeric expression would be appropriate. The rules for operators with textual values are as follows:

Arithmetic operators (+, −, *, /, ^ or **)

Textual values may not be used with arithmetic operators.

Relational operators (=, <>, <, >, <=, >=)

Textual values may be used to create expressions using the relational operators equal and not equal (=, <>).

Textual values may not be used with the remaining relational operators (<, >, <=, >=).

Valid relational expressions involving textual operands may appear in more complex expressions, such as logical function references.

The IF function may contain expressions with textual values as the second (*true case*) or third (*false case*) parameters.

Lookup tables may be constructed using expressions containing textual values.

IF Functions

- Calendar functions — The value of a calendar function may be used in the comparison expression.

 IF(B20 = DATE(02,25,47),Expression2,Expression3)

 If the value of B20 is the date 2/25/47, use expression 2, otherwise use expression 3.

- **("Text")** — A textual value may be used in a comparison expression.

 IF(A1 = ("Debit"),Expression2,Expression3)

 If the value of A1 is the textual value Debit, use expression 2, otherwise, use expression 3.



Learning to Use SuperCalc3